**Final Report**

**on**

**Quick Billing System**

**to be developed to fulfill the requirements for**

**Major Project (CA133)**

**Submitted to**

**Department of Computer Applications**

**Chitkara University, Punjab**



**under the supervision of**

**Dr. Preetinder Singh Brar**

**Associate Professor**

**Dr. Neelam Dahiya**

**Assistant Professor**

**Submitted by:-**

**Chetan Aashutosh Naveen**

**2110992712 2110992707 2110992741**

**Bachelor of Computer Applications**

**(Batch 2021-24)**

# CERTIFICATE

This is to certify that the report on software project titled “Quick Billing System” submitted by the student team comprising Chetan (2110992712), Aashutosh (2110992707), Naveen (2010992741) to the Department of Computer Applications, Chitkara University, Punjab in partial fulfillment for the completion of the course Major Project (CA133) in the fifth semester of Bachelor of Computer Applications is a Bona fide record of work carried out by the team under my supervision.

**Date:** Dr. Preetinder Brar

Associate Professor

Dr. Neelam Dahiya

Associate Professor

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
| **1.** | **Abstract of the project** | 1 |
| **2.** | **Profile of problems assigned** | 1-2 |
| **3.** | **Study of existing system** | 2 |
| **4.** | **System requirements**   * Product Definition   + Problem Statement   + Function to be provided   + Processing Environment   + Solution Strategy   + Acceptance Criteria * Feasibility Analysis * Project Plan   + Team Structure   + Development Schedule   + Programming Languages and Development Tools | 2-5 |
| **5.** | **System Requirement Specifications (SRS)**   * Developing / Operating Environments * External Interface and Data Flows   + User Display and report format, user command summary   + High level DFD and data dictionary * Functional and performance specifications | 5 |
| **6.** | **Design**   * Detailed DFD’s and structure diagrams * Data structures , database and file specifications. * Pseudo Code | 6-29 |
| **7.** | **Test Plan (in confirmation with the SRS)**   * Functional, Performance , Stress tests etc. | 29-39 |
| **8.** | **Implementation/ Conversion Plan** | 40 |
| **9.** | **Project Legacy**   * Current status of the project * Remaining areas of concern * Technical and managerial lessons learn * Future recommendation |  |
| **10.** | **Bibliography** | 40 |
| **11.** | **Source Code** | 40-51 |

**1.Abstract**

The Billing System Project is a strategic initiative aimed at revolutionizing the way shopkeepers manage their purchase and sales records. In response to the challenges posed by manual record-keeping, which is not only time-consuming but also prone to errors, our automated billing system emerges as the solution.

The key objectives of this project are to streamline shopkeepers' processes, ensuring they maintain accurate records and, more importantly, gain insightful statistical overviews of their business performance. By implementing this system, we anticipate a significant reduction in the administrative burden on shopkeepers, accompanied by a marked improvement in data accuracy. This, in turn, will empower better decision-making, enhancing overall operational efficiency and productivity.

The primary beneficiaries of the Billing System Project are shopkeepers across diverse industries and scales, ranging from small enterprises to large-scale businesses. Its application extends to retail stores, grocery shops, pharmacies, and various other establishments. What sets this system apart is its commitment to inclusivity, catering to the needs of shopkeepers with varying technical expertise.

The system boasts a user-friendly interface and customizable features, ensuring accessibility for all. Its adaptability makes it a valuable tool for shopkeepers seeking efficiency without the burden of complex technology. Furthermore, the project extends its scope to encompass the educational realm, welcoming college students and developers interested in delving into the intricacies of web development, database management, and the practical implementation of real-world applications.

In essence, the Billing System Project not only addresses immediate business challenges but also serves as a gateway for individuals keen on expanding their knowledge in relevant technological domains. It stands as a beacon of professionalism and efficiency, promising a seamless and attractive solution for the dynamic landscape of shop management.

**2.Profile of problems assigned**

Manual transactions lead to prolonged invoice generation times, occasional errors in billing calculations, and an overall suboptimal user experience. This inefficiency has resulted in increased customer complaints, delayed payments, and a negative impact on the company's reputation. Additionally, the system does not adequately accommodate varying discount structures, making it challenging to implement diverse pricing strategies for different customer segments. As a result, the organization is facing operational inefficiencies, decreased customer satisfaction, and potential revenue loss due to billing inaccuracies. This problem statement identifies the key issues within the billing system website, including transaction processing inefficiencies, billing errors, negative customer impact, and limitations in handling diverse discount structures. It serves as a foundation for addressing these specific challenges and improving the overall functionality and performance of the billing system.

**3.Study of existing System**

The current manual billing system employed by shopkeepers involves a traditional approach to recording purchase and sales transactions. In this system, shopkeepers manually document details such as product names, quantities, prices, and customer information on paper or in physical ledgers. The process is time-consuming and prone to errors, relying heavily on manual data entry and calculations.

Challenges and Limitations:

1. Data Handling and Accuracy:

Data accuracy is a significant concern due to the reliance on manual input.

The potential for errors in recording product details, quantities, and customer information poses a risk to overall data integrity.

2. Time and Resource Constraints:

The manual nature of the system demands considerable time and human resources.

Shopkeepers often find themselves spending excessive time on record-keeping tasks, diverting energy from other business priorities.

3. Lack of Analytical Insights:

The current system lacks the capability to generate insightful statistical overviews.

Business intelligence is compromised as the system does not provide comprehensive reports or analytical tools.

4. User Feedback and Pain Points:

Feedback from shopkeepers highlights frustration with manual data entry and the time required for maintaining records.

Common pain points include difficulties in retrieving historical data and the absence of a user-friendly interface.

**4.System Requirement**

**a)Problem Statement**

The Billing System website addresses a pressing issue faced by shopkeepers in the manual management of purchase and sales records. Currently, the conventional method of record-keeping is marred by inefficiencies, time consumption, and a susceptibility to errors. Shopkeepers, irrespective of the scale or nature of their businesses, encounter challenges in tracking the details of their transactions, including the quantity and types of items purchased, distinguishing between tax and non-tax bills, and generating insightful statistical overviews of their billing history. The existing manual processes not only consume valuable time but also introduce a significant margin for error, jeopardizing the accuracy of financial records. Shopkeepers often find themselves burdened with administrative tasks, diverting precious resources away from core business activities. Furthermore, the lack of a centralized and automated system hampers the ability to extract meaningful insights from the data accumulated over time.

The Billing System project seeks to address these challenges by offering an integrated platform where shopkeepers can effortlessly log in, input essential details such as shop name, and record their bills in a systematic manner. The platform's user-friendly interface ensures accessibility for all, enabling shopkeepers to manage their records efficiently. By automating the billing process and providing statistical overviews, the system alleviates the burden of manual record-keeping, enhances data accuracy, and empowers shopkeepers to make informed decisions for the growth of their businesses.

**b) Function to be provided**

A billing system website typically serves as a crucial component for managing financial transactions, invoicing, and related processes. The functions provided in a billing system website may vary based on business needs, industry requirements, and the complexity of the billing processes. Here are essential functions that are commonly found in billing system websites:

1. User Authentication and Authorization:

- Secure login mechanisms for both customers and administrators.

- Role-based access control to manage permissions for different user roles.

2. Customer Management:

- Create and manage customer profiles with relevant information.

- Track customer history, including transactions, invoices, and payment records.

3. Invoice Generation:

- Generate accurate and detailed invoices based on transactions.

- Customizable invoice templates with company branding and information.

4. Product Catalog:

- Maintain a catalog of products including descriptions and pricing.

- Support for adding, updating, and removing items from the catalog.

5. Transaction Processing:

- Process transactions related to product purchases.

- Automatic calculation of transaction amounts based on quantity and price.

**c)Processing Environment: Hardware & Software**

|  |  |
| --- | --- |
| Client Side Operating System :- |  |
| 1. Processor:  A modern processor with at least a dual-core CPU, such as Intel Core i3 or AMD Ryzen 3, to ensure smooth handling of website tasks and interactivity.  2. Operating System:  The website should be compatible with commonly used operating systems, including Windows, macOS, and Linux, to cater to a wide range of users.  3. Memory (RAM):  A minimum of 4GB RAM is recommended for optimal performance. Higher RAM capacity will provide a better browsing experience, especially when handling multiple browser tabs simultaneously.  4. Hard Disk Space:  The website itself doesn't require significant hard disk space, but a sufficient amount of available storage is necessary to accommodate temporary files and the browser cache.  5. Software Version:  The website should be compatible with popular web browsers and their recent versions, such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. This ensures proper rendering of the website's design and functionalities. | |

**Server Side:-**

|  |  |
| --- | --- |
| **Processor** | A modern processor with at least dual-core capabilities, such as entry level Intel i3 or equivalent AMD processor, will suffice for handling the limited number of request. |
| **Operating system** | You can choose a lightweight and user-friendly operating system like Linux (e.g., Ubuntu) or Windows (e.g., Windows Server) for hosting your website. |
| **Memory** | For a small-scale website, 1GB or 2GB of RAM should be sufficient to handle concurrent requests from a limited number of users effectively. |
| **Hard disk space** | Minimum 128GB and recommended 256 GB or more. |

**d) Solution Strategy**

1. Problem Analysis and Stakeholder Input:

- Conduct a thorough analysis of the current billing system, identifying pain points and areas for improvement.

- Gather input from stakeholders, including users, administrators, and decision-makers, to understand their perspectives on system shortcomings.

2. Technology Stack Evaluation:

- Evaluate the existing technology stack for suitability and scalability.

- Consider modern technologies and frameworks that align with the system's requirements.

3. Transaction Processing Optimization:

- Streamline transaction processing to reduce latency and enhance system responsiveness.

- Optimize database queries and indexing for efficient data retrieval.

4. Security Enhancements:

- Implement robust security measures, including data encryption, secure user authentication, and authorization.

- Conduct regular security audits to identify and address vulnerabilities.

5. Scalability Planning:

- Design the system architecture with scalability in mind to handle increasing transaction volumes.

- Consider cloud-based solutions for flexibility and scalability.

**e.) Acceptance Criteria**

The acceptance criteria for the Billing System website serve as measurable benchmarks to ensure that the developed solution meets the expectations and requirements of shopkeepers. These criteria encompass various aspects of functionality, usability, and performance, reflecting the desired outcomes of the project. Here are the acceptance criteria for the Billing System website:

1. User Registration and Profile Management:

- Criteria: Users should be able to register on the platform seamlessly.

- Measurement: Successful completion of the registration process without errors. The user profile should be created and managed effectively.

2. Shop Information Setup:

- Criteria: Shopkeepers should be able to input and modify their shop information.

- Measurement: Completion of the setup process without system errors. Ability to customize and update shop details successfully.

3. Billing Record Entry**:**

- Criteria: Users should be able to enter purchase and sales records accurately.

- Measurement: Error-free entry of billing details. The system should correctly distinguish between tax and non-tax bills.

4. Automated Calculation and Tax Management:

**-** Criteria: The system should accurately perform automated calculations and manage tax information.

- Measurement: Verification that the system provides correct total bills and appropriately handles tax-related calculations.

5. Data Security and Backup:

- Criteria: The system should prioritize data security and conduct regular backups.

- Measurement: Verification of implemented security measures, and confirmation of successful data backups at predefined intervals.

6. Overall System Performance:

- Criteria: The system should perform efficiently and without significant delays.

- Measurement: Verification of system responsiveness and absence of critical performance issues during regular usage.

**b. Feasibility Analysis**

Feasibility analysis is a crucial step in determining whether a billing system website is viable and worth pursuing. It typically involves assessing technical, operational, economic, legal, and scheduling aspects of the project. Here's a breakdown of key considerations for a billing system website:

1. Technical Feasibility:

- Hardware and Software Requirements: Assess the technical requirements for servers, databases, and other infrastructure components.

- Development Tools: Evaluate the availability and suitability of programming languages, frameworks, and other development tools.

- Integration: Consider the ability to integrate with existing systems, databases, and third-party services.

2. Operational Feasibility:

- User Requirements: Identify and understand the needs of end-users, administrators, and other stakeholders.

- Training and Support: Assess the feasibility of training users and providing ongoing support for the system.

- Change Management: Evaluate the impact of the new system on existing business processes and workflows.

3. Economic Feasibility:

- Cost-Benefit Analysis: Estimate the costs associated with development, implementation, maintenance, and training, and compare them with the expected benefits.

- Return on Investment (ROI): Analyze the potential financial gains and the time it will take to recover the initial investment.

4. Security Feasibility:

- Data Security: Evaluate measures to ensure the security and integrity of sensitive customer and financial data.

- Authentication and Authorization: Implement robust user authentication and authorization mechanisms to control access to sensitive information.

5. Usability:

- User Interface: Ensure that the billing system has an intuitive and user-friendly interface to enhance user adoption and satisfaction.

**c. Project plan**

**a)Team Structure**

The team structure for the development of the Billing System website involves a collaborative effort among three key contributors: Naveen, Ashutosh, and Chetan. Each team member plays a distinct role, contributing their expertise to ensure the successful creation of a comprehensive and functional website.

1. Ashutosh (Content and Backend):

- Role: Ashutosh takes on a dual role, focusing on both content creation and backend development. In terms of content, Ashutosh is responsible for crafting the textual elements of the website, ensuring clarity, coherence, and alignment with the project's goals. Simultaneously, in the backend development realm, Ashutosh contributes to the architecture, data management, and overall functionality of the website.

2. Naveen (Frontend):

- Role: Naveen is the key contributor to the frontend development of the Billing System website. His primary focus is on creating an intuitive and visually appealing user interface. Naveen translates design concepts into interactive elements, ensuring a seamless and engaging user experience. His work involves implementing responsive design, incorporating user feedback, and optimizing the website for various devices.

3. Chetan (Frontend & Backend):

- Role: Chetan is dedicated to the backend development of the Billing System website. His responsibilities include designing and implementing the server-side logic, database management, and ensuring the overall functionality of the website. Chetan works on creating a robust and efficient backend architecture that supports the seamless processing of data, user authentication, and other essential functionalities.

**b)Development Schedule**

Phase 1: Project Planning and Design (Week 1-4)

-Week 1-2: Project Kickoff and Role Assignment

Form project team.

Conduct project initiation meeting.

Define project roles and responsibilities.

-Week 3-4: Feature Planning and Design

Discuss and conclude on website features.

Outline how the website will function.

Phase 2: Frontend Development (Week 5-12)

-Week 5-8: Homepage Frontend Development

Design wireframes for the homepage.

Begin frontend development of the homepage.

Implement features such as navigation, headers, and footers.

-Week 9-12: Other Pages and Features

Complete frontend development of other pages (About, FAQ, etc.).

Integrate additional features on the homepage.

Ensure responsive design for various devices.

Phase 3: Admin and User Page Development (Week 13-20)

-Week 13-16: Admin Page Frontend Development

Design wireframes for the admin page.

Begin frontend development of the admin page.

Implement features for admin functionalities.

-Week 17-20: User Page Frontend Development

Design wireframes for the user page.

Begin frontend development of the user page.

Implement features for user interactions.

Phase 4: Backend and Database Development (Week 21-28)

-Week 21-24: Backend Development

Implement server-side logic for admin and user functionalities.

Establish communication between frontend and backend.

Week 25-28: Database Setup and Integration

Design and set up the database structure.

Integrate the frontend with the database.

Perform thorough testing of data flow.

Phase 5: Testing and Optimization (Week 29-34)

Week 29-32: Testing

Conduct unit testing for each component.

Perform integration testing for seamless interactions.

Address any bugs or issues identified.

Week 33-34: Optimization and Final Adjustments

Optimize system performance.

Make final adjustments based on testing feedback.

Prepare for User Acceptance Testing (UAT).

**c)Programming languages and Development tools**

The development of the Billing System website involves the strategic utilization of specific programming languages and development tools to ensure a robust and efficient system. Here is a breakdown of the programming languages and development tools employed in both frontend and backend development:

Frontend Development: HTML and CSS

1. HTML (Hypertext Markup Language):

HTML is the backbone of the website's structure, defining the elements and layout of web pages. It structures content, incorporating headers, paragraphs, images, and other essential elements.

2. CSS (Cascading Style Sheets):

CSS complements HTML by styling and formatting the website's visual presentation. It controls the layout, color schemes, fonts, and overall aesthetic appeal, ensuring a cohesive and visually pleasing user interface.

3. Development Tools for Frontend:

Tools like Visual Studio Code, Sublime Text, or Atom are utilized for code editing, providing features like syntax highlighting and autocompletion.

Browser Developer Tools: Browsers like Chrome, Firefox, or Safari offer built-in developer tools that aid in testing, debugging, and optimizing frontend code.

Version Control: Git is employed for version control, allowing collaborative development and tracking changes to the codebase.

Backend Development: JavaScript, PHP, and MySQL

1. JavaScript:

JavaScript enhances the interactivity of the Billing System website. It is used to implement dynamic features, handle user input, and facilitate seamless communication between the frontend and backend.

2. PHP (Hypertext Preprocessor):

PHP is the server-side scripting language responsible for processing backend logic. It handles tasks such as user authentication, database interactions, and server-side computations, ensuring the website's functionality.

3. MySQL:

MySQL serves as the relational database management system (RDBMS) for the Billing System. It stores and retrieves data efficiently, supporting tasks such as user authentication, storing purchase and sales records, and generating statistical overviews.

4. Development Tools for Backend:

Server Environment: Tools like XAMPP, WampServer, or MAMP provide a local server environment for developing and testing PHP and MySQL applications.

Database Management Tools: PHPMyAdmin or MySQL Workbench are utilized for database management, allowing efficient handling of database schema, queries, and data.

Code Editors: Similar to frontend development, code editors such as Visual Studio Code or Sublime Text are used for PHP coding, offering features for syntax highlighting and code completion.

**5. Software Requirement Specification**

**a) Developing / Operating/ Maintenance Environments**

**a. Developmental Environment:**

1. Programming Languages:

- Frontend development utilizes HTML and CSS for structure and style, while JavaScript enhances interactivity.

- Backend development is carried out using PHP for server-side scripting, and MySQL serves as the relational database management system.

2. Frameworks and Libraries:

- The frontend development leverages common frameworks such as Bootstrap or equivalent for responsive design.

3. Development Tools:

- Integrated Development Environments (IDEs) such as Visual Studio Code facilitate code editing and debugging.

- Local server environments like XAMPP or WampServer aid in testing PHP and MySQL applications during development.

**b. Operating Environment:**

1. Web Browsers:

- The Billing System website is optimized for major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

- Cross-browser compatibility ensures a consistent user experience.

2. Operating Systems:

- The system is designed to operate seamlessly on multiple operating systems, including Windows, macOS, and Linux.

- Cross-platform compatibility enhances accessibility for a diverse user base.

3. Internet Connection:

- A stable internet connection is required for users to access and interact with the website.

- The system is optimized for both broadband and mobile internet connectivity.

4. Device Compatibility:

- The website is responsive and compatible with various devices, including desktops, laptops, tablets, and smartphones.

- Responsive design ensures a user-friendly experience on screens of different sizes.

**c. Maintenance Environment:**

1. Software Updates:

- Regular updates are released to enhance features, improve security, and address any bugs.

- Users are notified of updates, and the update process is designed to be user-friendly.

2. Database Management:

- Routine database maintenance involves checks, optimization, and backup procedures.

- Database management tools like PHPMyAdmin or MySQL Workbench aid in efficient administration.

3. Server Environment:

- The server environment requires regular monitoring, security updates, and performance optimization.

- Routine server maintenance tasks are performed to ensure optimal performance and reliability.

5. User Support and Documentation:

- A user support system is in place to address queries and issues promptly.

- Comprehensive documentation, including user guides and troubleshooting resources, is available to assist users and the development team.

6. Backup Procedures:

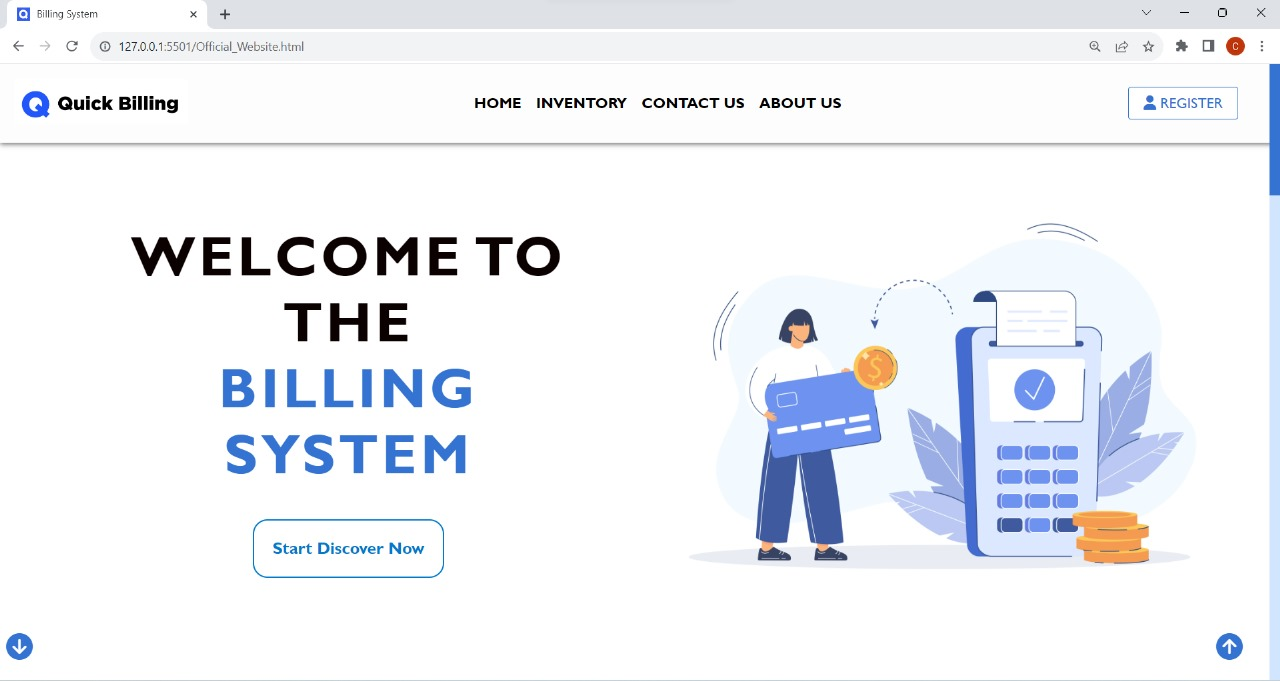
- Regular automated backups of the database and essential system files are performed to prevent data loss.

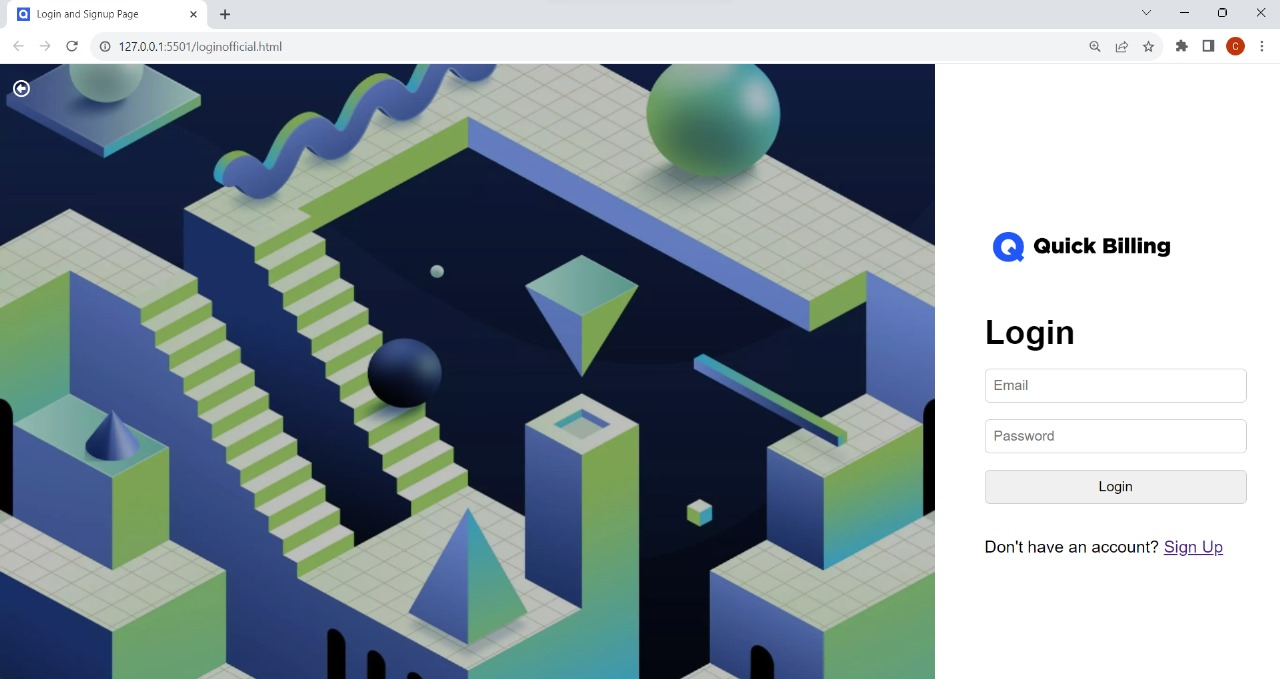
- Backup procedures are periodically tested to ensure data recovery readiness.

**b. External Interface and Data Flows**

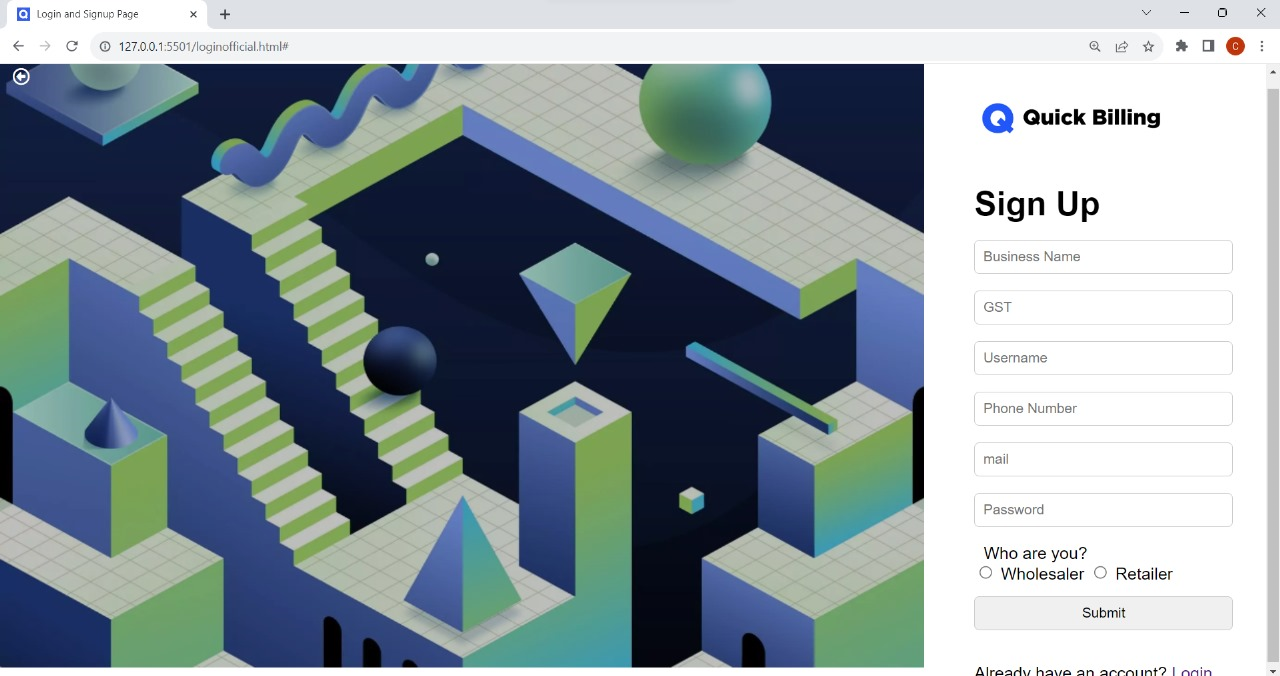
**a) User display-**

**1. Home Page of Quick Billing**

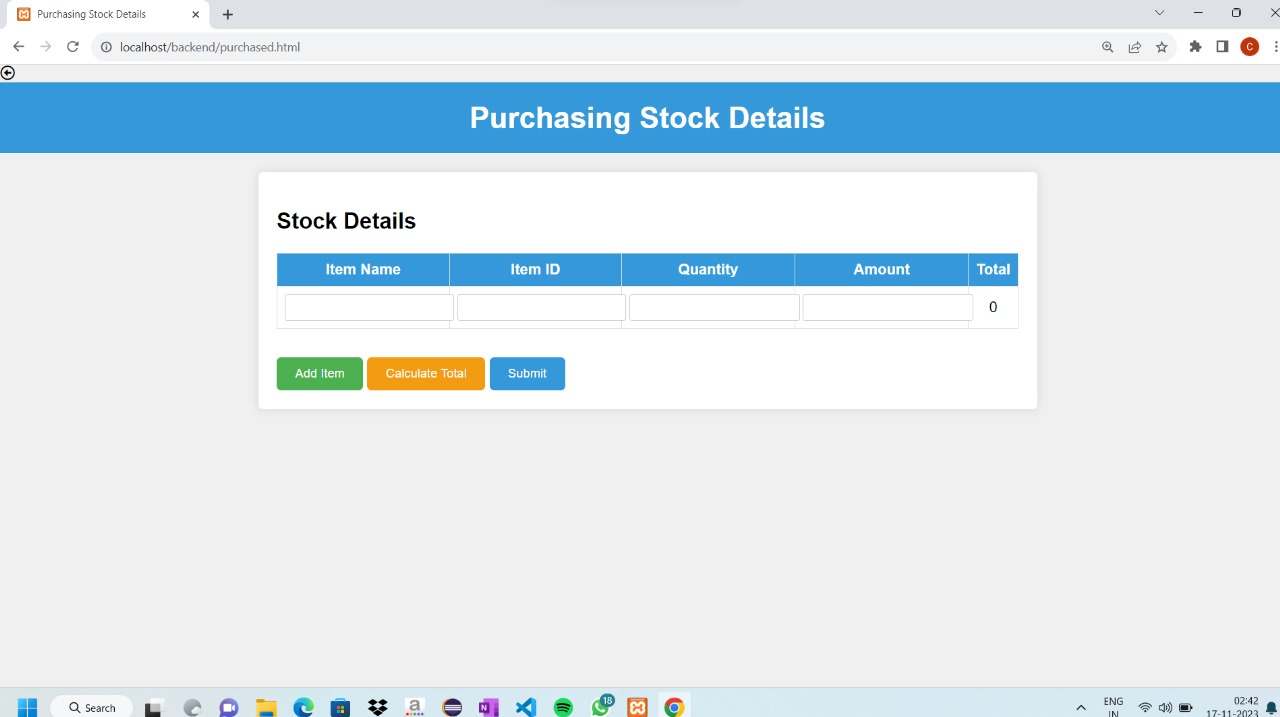




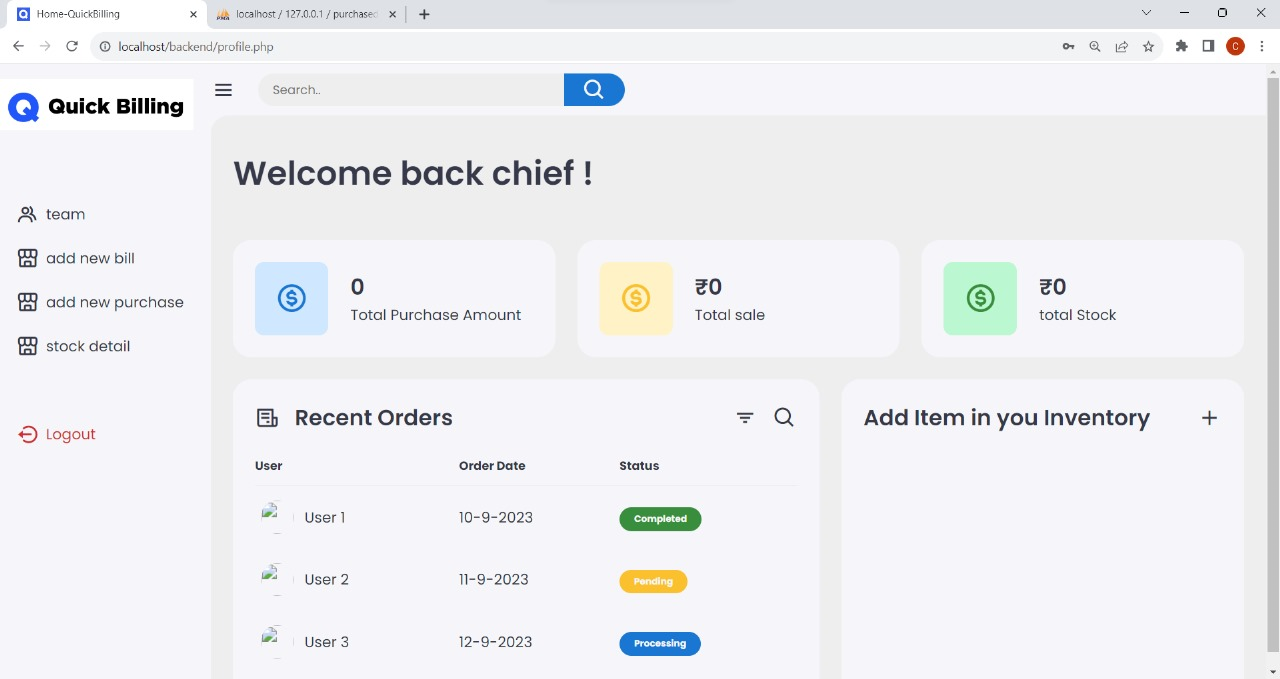
**2. Login Page**



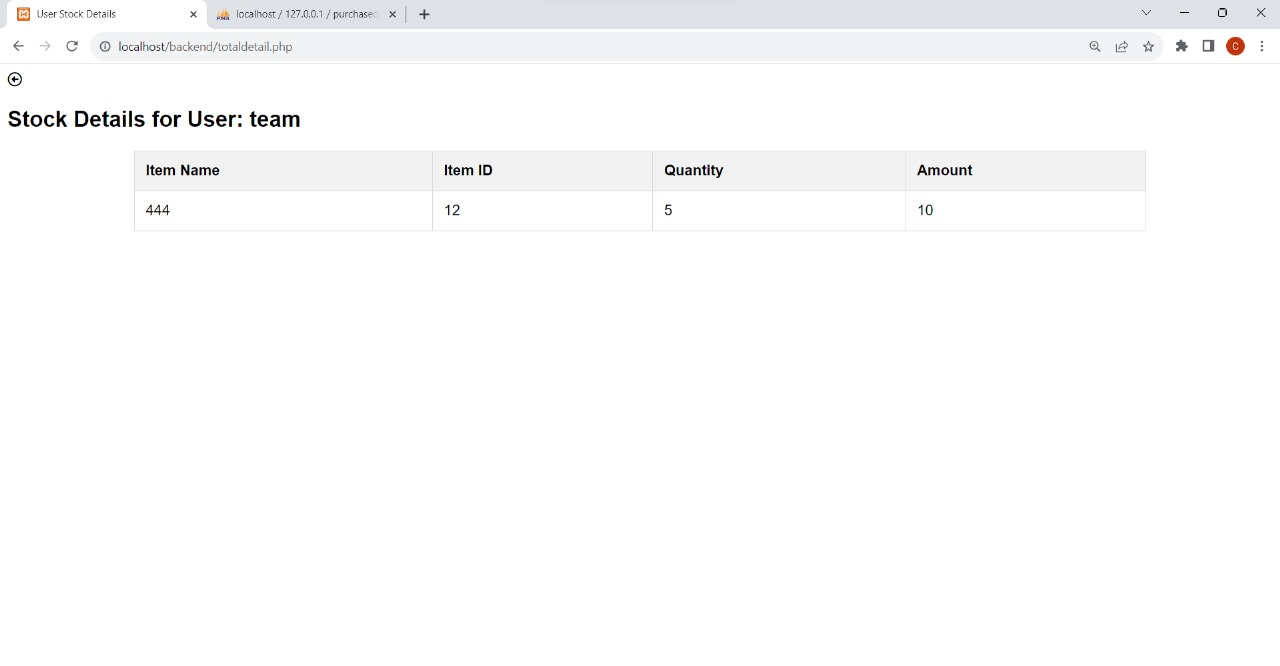
**3. Sign Up Page**



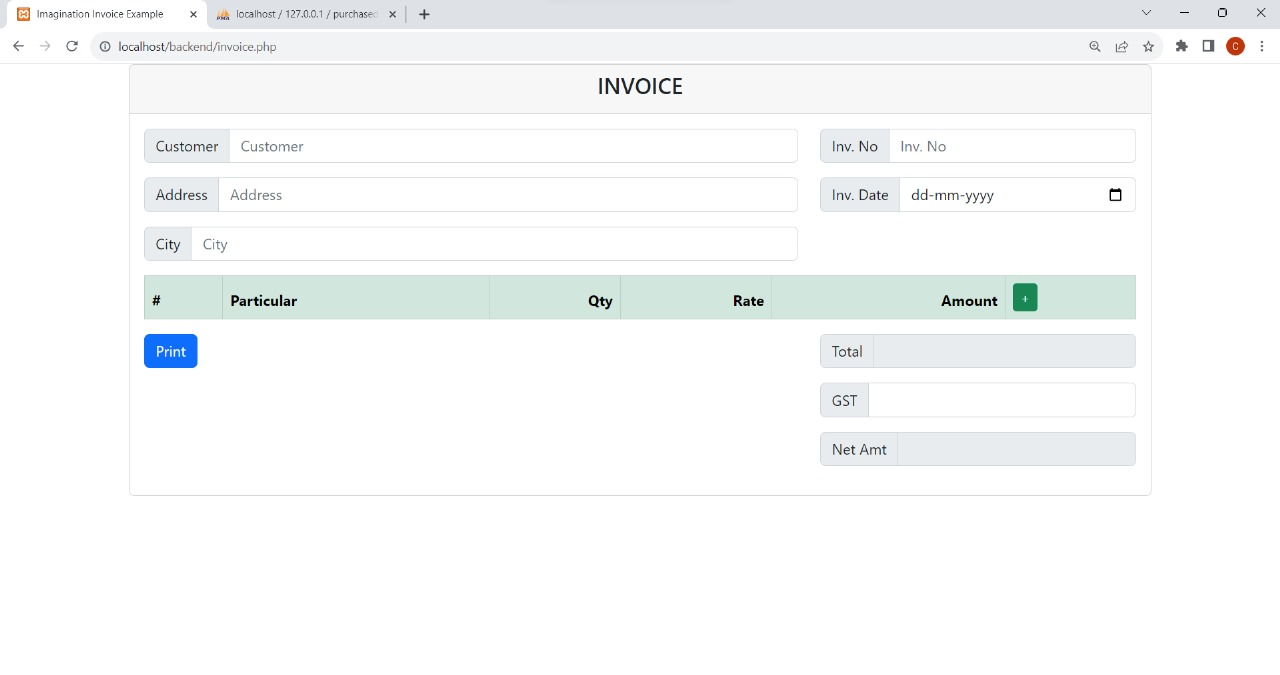
**4. Stock Entry Page**



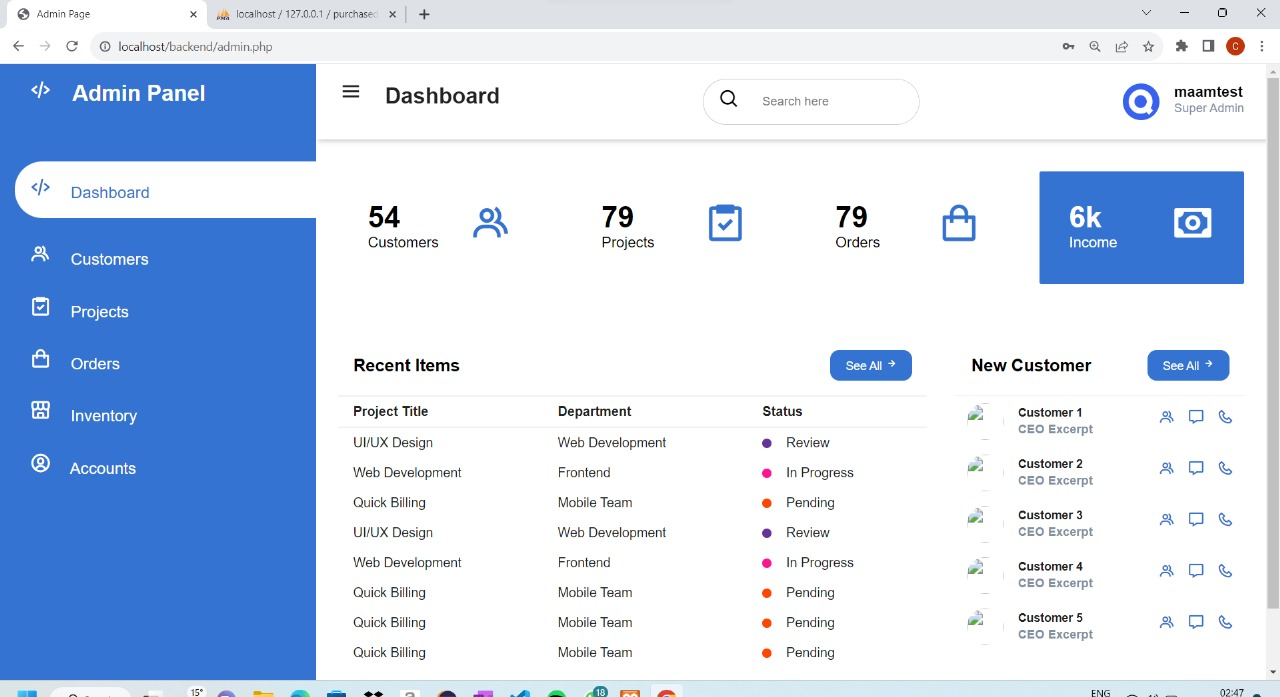
**5. Dashboard**



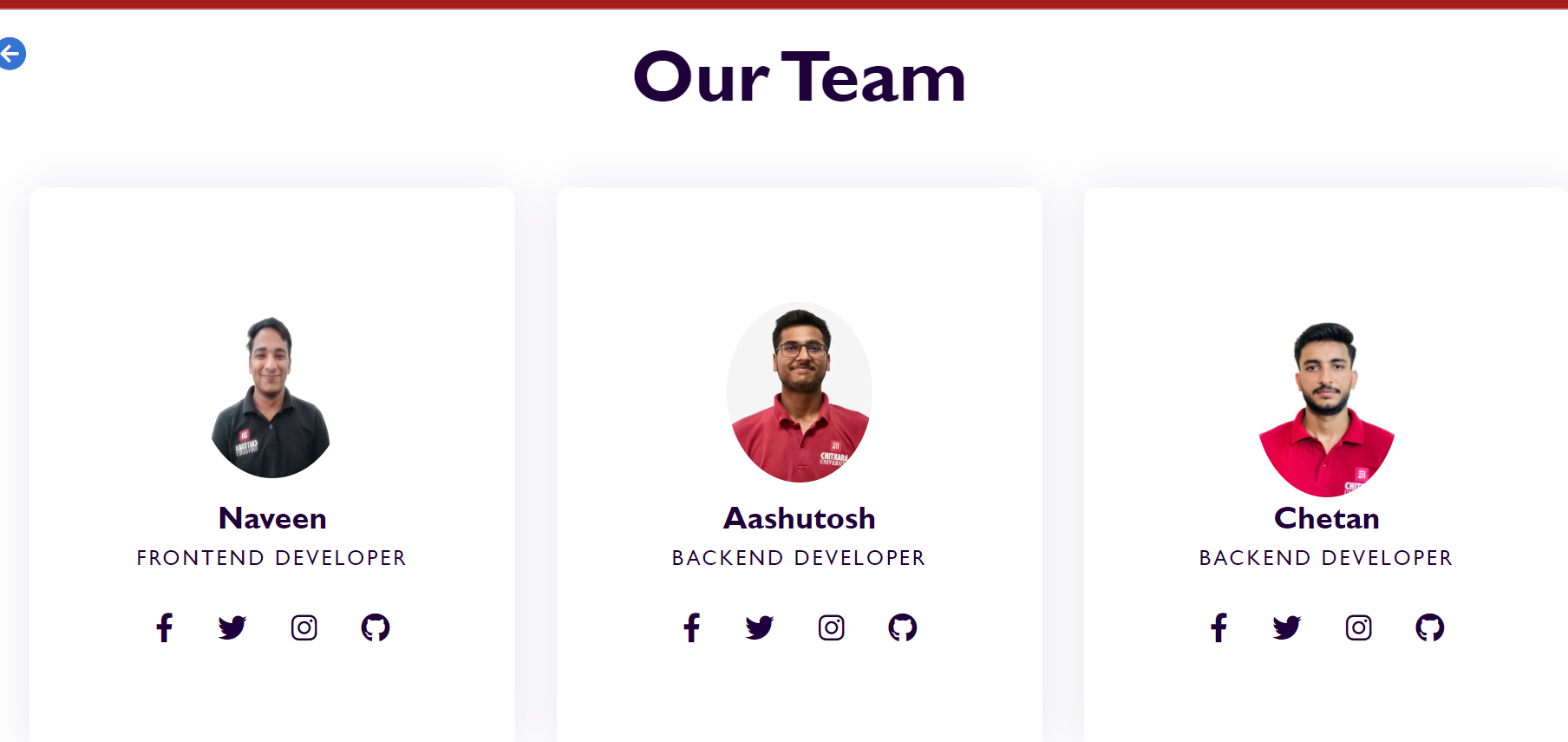
**6. Stock Display Page**



**7. Invoice Genrator**



**8. Admin Panel**

****

**9. Team Page**

**4.b.(a) Report Format**

1. Invoice Report:

- Structure: The invoice report is presented in a professional format, including:

- Header: Shop name, logo, and contact details.

- Transaction Details: Date, items purchased, quantities, prices, and total amount.

- Customer Details (if applicable): Name, contact information.

- Footer: Thank-you message and any additional notes.

2. Statistical Overview Report:

- Structure: The statistical overview report provides insights into sales trends and business performance, featuring:

- Graphs and Charts: Visual representations of sales trends over time.

- Key Metrics: Total sales, top-selling items, and peak sales periods.

- Comparisons: Month-to-month or year-over-year comparisons for strategic analysis.

3. User Profile Report:

- Structure: The user profile report summarizes the shopkeeper's information, displaying:

- Profile Details: Name, shop name, contact details.

- Account Information: Email address, username.

- Recent Activity: Log of recent transactions and interactions with the system.

4. Forget Password Request Report:

- Structure: The report related to forget password requests ensures security and includes:

- Request Details: Timestamp of the forget password request.

- Email Verification Status: Confirmation of whether the email verification was successful.

- Reset Status: Confirmation of whether the password was successfully reset.

**4.b.(a) User command Summary**

1. User Registration:

- Command: Users initiate the registration process by providing their name, phone number, and email address.

- Summary: The registration process creates a user profile, allowing shopkeepers to access personalized features and record their purchase and sales information.

2. Shop Information Setup:

- Command: Shopkeepers input details such as the shop name and type during the initial setup.

- Summary: Setting up shop information customizes the platform to cater to the specific needs of the business, ensuring accurate record-keeping.

3. Billing Record Entry:

- Command: Shopkeepers enter details of purchase and sales transactions, including item quantity and type.

- Summary: This command records crucial billing information, facilitating accurate financial tracking and analysis.

4. Automated Calculation and Tax Management:

- Command: The system automatically calculates total bills and manages tax-related information.

- Summary: Automated calculations reduce errors, ensuring precise billing and streamlined tax management for shopkeepers.

5. Statistical Overview and Reporting:

- Command: Shopkeepers access statistical overviews of their billing history and generate reports.

- Summary: Statistical overviews offer insights into sales trends, helping shopkeepers make informed decisions for business growth.

6. User Profile Management:

- Command: Users can manage their profiles, updating information such as contact details.

- Summary: Profile management ensures up-to-date information for effective communication and personalized user experience.

7. Regular Updates and Maintenance:

- Command: Users are informed of system updates, and maintenance is conducted regularly.

- Summary: Regular updates enhance features, improve security, and ensure the system's optimal performance.

8. Data Backup and Security Checks:

- Command: Automated backups of data and periodic security checks are conducted.

- Summary: These commands ensure data integrity, security, and quick recovery in case of unforeseen events.

9. Generate Invoice:

- Command: Shopkeepers can generate invoices for their transactions by selecting the respective option in the dashboard.

- Summary: Invoice generation compiles the details of a specific transaction into a formal document, aiding in financial record-keeping and customer communication.

10. Forget Password - Step 1: Request Password Reset:

- Command: Users who forget their passwords can initiate a reset by selecting the "Forgot Password" option on the login page.

- Summary: This initiates the password reset process, ensuring secure access to the account in case of forgotten credentials.

11. Forget Password - Step 2: Email Verification:

- Command: An email is sent to the user's registered email address with a unique verification link.

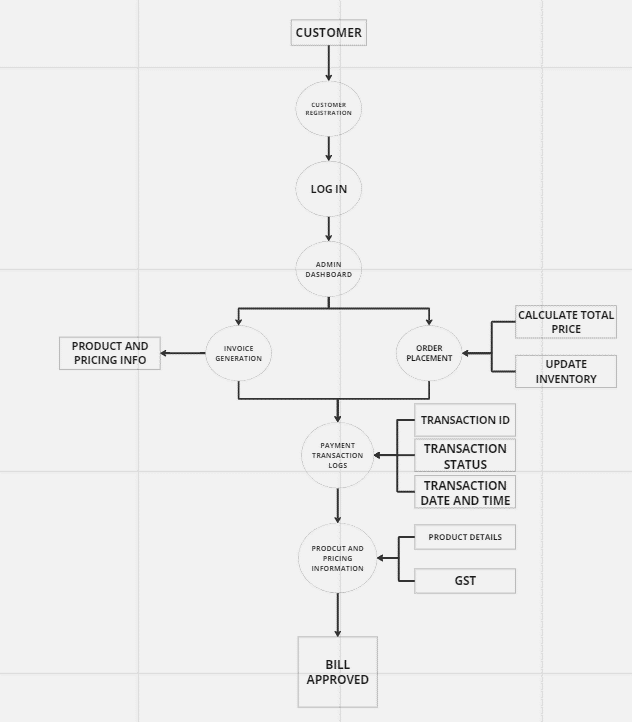
- Summary: Email verification ensures the authenticity of the password reset request and provides a secure means of resetting the password.

12. Forget Password - Step 3: Reset Password:

- Command: Users click on the verification link, which directs them to a secure page to reset their password.

- Summary: This step allows users to set a new password, completing the forget password process and ensuring account security.

**8.b.(b)High Level DFD and data dictionary**



**1.Working of Quick Billing System**

**-Explanation of Working of Quick Billing System**

1.User Registration and Login

Description: Allows users to register and log in to the system.

Inputs:

User information (username, password, email, phone number, shop details).

Outputs:

User authentication status.

2. Bill Recording

Description: Enables users to record purchase and sales bills.

Inputs:

Bill details (items, quantity, total price, tax information).

Outputs:

Updated user and stock information.

3. View Bills and Stock Details

Description: Provides users with the ability to view recorded bills and stock details.

Inputs:

User request for bill or stock information.

Outputs:

Display of requested information.

4. Admin Functions

Description: Admin-specific functionalities, such as user management and system configuration.

Inputs:

Admin commands and configurations.

Outputs:

Updates to user and system settings.

Data Stores:

1. User Information

Description: Stores details about registered users.

Contents:

User details (username, password, email, phone number, shop details).

2. Bill Information

Description: Stores details about recorded bills.

Contents:

Bill details (bill ID, user ID, items, quantity, total price, tax information).

3. Stock Details

Description: Stores information about the stock of items.

Contents:

Item details (item ID, user ID, item name, quantity available, unit price).

External Entities:

1. Users

Description: Individuals interacting with the system.

Inputs:

Registration and login details.

Outputs:

Access to system features.

2. Admin

Description: System administrator managing user accounts and system settings.

Inputs:

Admin commands and configurations.

Outputs:

System updates and configurations.

**8.b.(b)Data Dictionary**

1. User Information Table:

1.1. user\_table

Description: Stores information about registered users.

Attributes:

user\_id (Primary Key): Unique identifier for each user.

username: User's chosen username for login.

password: User's password for login (encrypted).

email: User's email address for communication.

phone\_number: User's contact number.

shop\_name: Name of the shop associated with the user.

shop\_type: Type of shop (e.g., retail, grocery).

registration\_date: Date when the user registered.

2. Bill Information Table:

2.1. bill\_table

Description: Stores details about bills recorded by users.

Attributes:

bill\_id (Primary Key): Unique identifier for each bill.

user\_id (Foreign Key): References the user\_table for associating bills with users.

bill\_date: Date when the bill was recorded.

items: List of items included in the bill.

quantity: Quantity of each item in the bill.

total\_price: Total price of the bill.

tax\_type: Indicates whether the bill includes tax or not.

payment\_status: Indicates if the bill has been paid.

3. Stock Details Table:

3.1. stock\_table

Description: Stores information about the stock of items.

Attributes:

item\_id (Primary Key): Unique identifier for each item.

user\_id (Foreign Key): References the user\_table to associate items with users.

item\_name: Name of the item.

quantity\_available: Quantity of the item available in stock.

unit\_price: Price of one unit of the item.

last\_updated: Date of the last update to the item's information.

**8.c) Functional and Performance specification**

Functional Specifications

1. User Registration and Authentication:

Functionality:

Users can register with their username, password, email, and phone number.

The system validates and stores user information securely.

Users can log in using their credentials.

Requirements:

User information should be stored securely in the database.

Passwords should be encrypted for security.

Successful login grants access to user-specific functionalities.

2. Bill Recording:

Functionality:

Users can record bills with details such as items, quantity, total price, and tax information.

The system updates user and stock information based on recorded bills.

Requirements:

User input should be validated for accuracy.

Recorded bills should be linked to the user's account.

Stock details should be updated accordingly.

3. View Bills and Stock Details:

Functionality:

Users can view their recorded bills and stock details.

Admins can access and manage all users' bills and stock information.

Requirements:

Access to bills and stock details should be restricted based on user roles.

Information should be presented in a clear and user-friendly format.

4. Admin Functions:

Functionality:

Admins can manage user accounts and configure system settings.

Admins have access to all user bills and stock details for oversight.

Requirements:

Admin functionalities should be password-protected.

Admins should be able to add, modify, or deactivate user accounts.

Performance Specifications

1. Response Time:

Requirement:

The system should respond to user interactions within 2 seconds.

Measurement:

Measure the time taken for the system to respond to user actions.

2. Data Retrieval Speed:

Requirement:

Retrieving bill and stock information should take less than 3 seconds.

Measurement:

Measure the time taken to fetch and display user and stock data.

3. Error Handling:

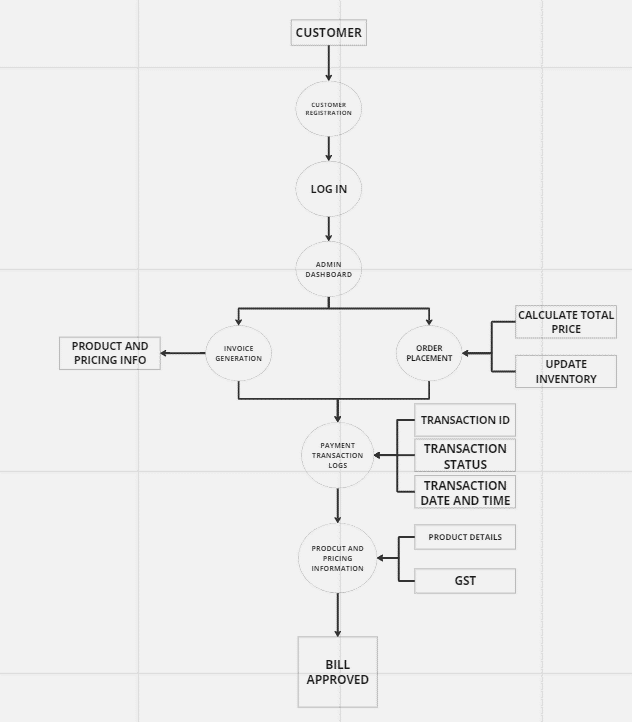
Requirement:

Error messages should be displayed within 5 seconds of an issue.

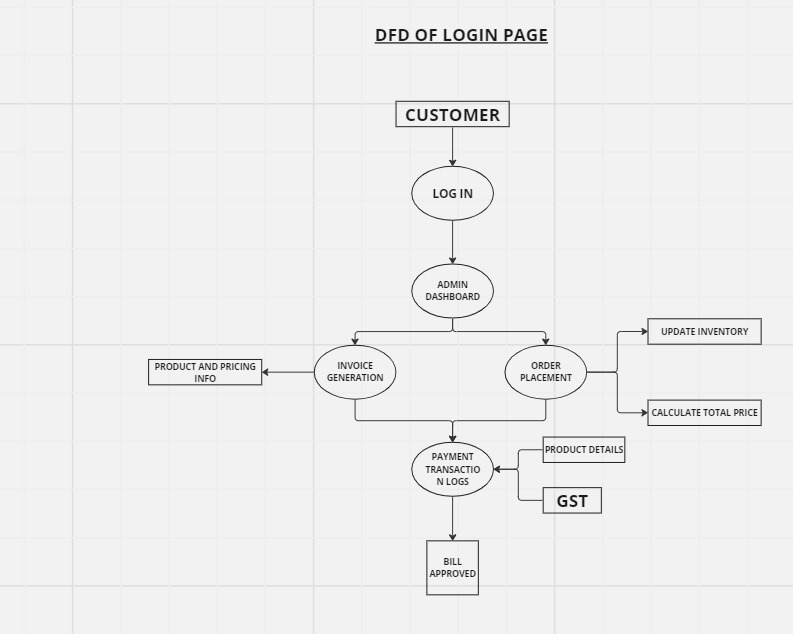
Measurement:

Test the system's ability to provide timely and informative error messages.

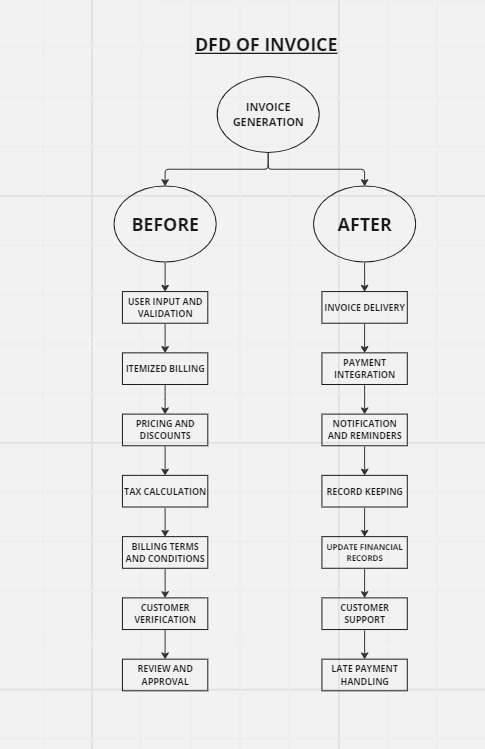
**6. a) Data Flow Diagrams (DFDs)**



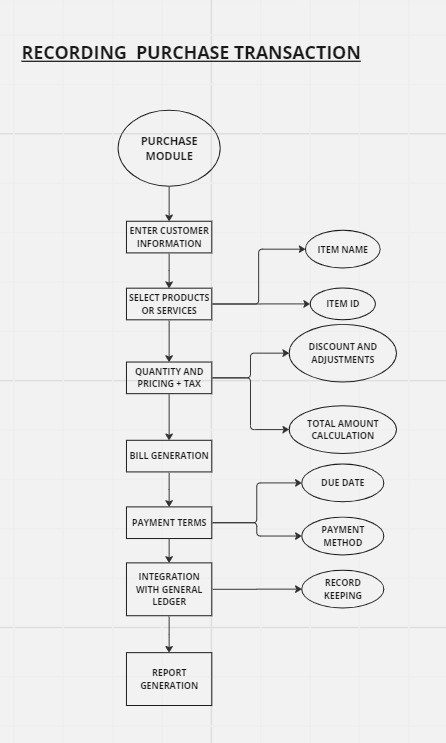
**1. Working of Quick Billing**



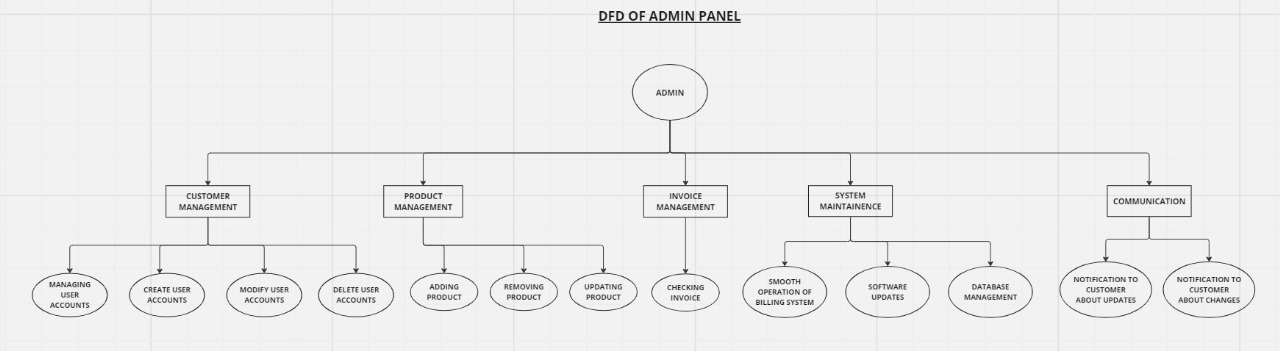
**2. Login Page**



**3.Invoice Generation**



**4. Recording Purchase Transaction**



**5. Admin Panel**

**9.b) Data Structure & File specification:**

1. Sign-Up Table:

Purpose:

This table, named signup\_table, is responsible for storing information related to user sign-up. It captures details about businesses or shopkeepers who register on your website.

Fields:

s\_no (Primary Key): An auto-incremented unique identifier for each sign-up entry.

business\_name: The name of the business or shop.

ph\_no: The phone number associated with the business.

gmail: The email address associated with the business.

username (Primary Key, Foreign Key): The username chosen by the user, serving as both the primary key and a foreign key linking to the stock\_details\_table.

password: The password chosen by the user for authentication.

gst\_no: The Goods and Services Tax (GST) number associated with the business.

2. Stock Details Table:

Purpose:

The stock\_details\_table is designed to store information about the stock or inventory associated with each user

Fields:

username (Primary Key): The username associated with the stock details, serving as both the primary key and a foreign key linking to the signup\_table.

item\_name: The name of the item or product in the stock.

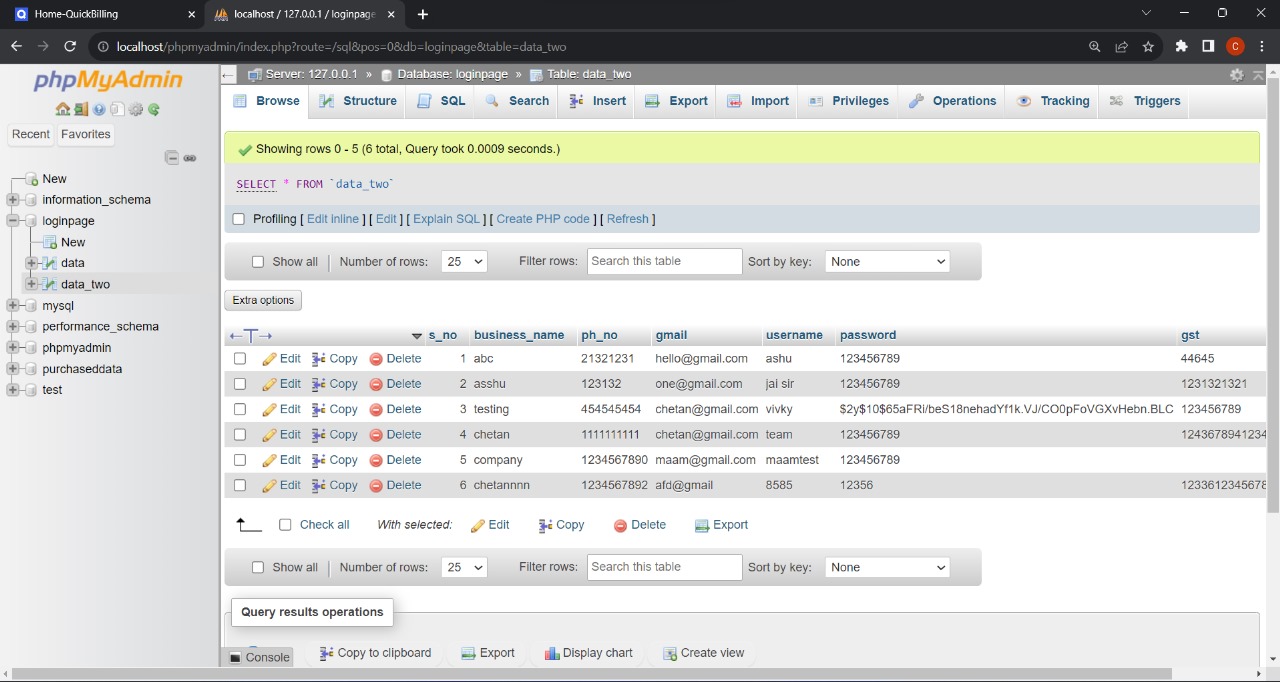
item\_id: An identifier for the item.

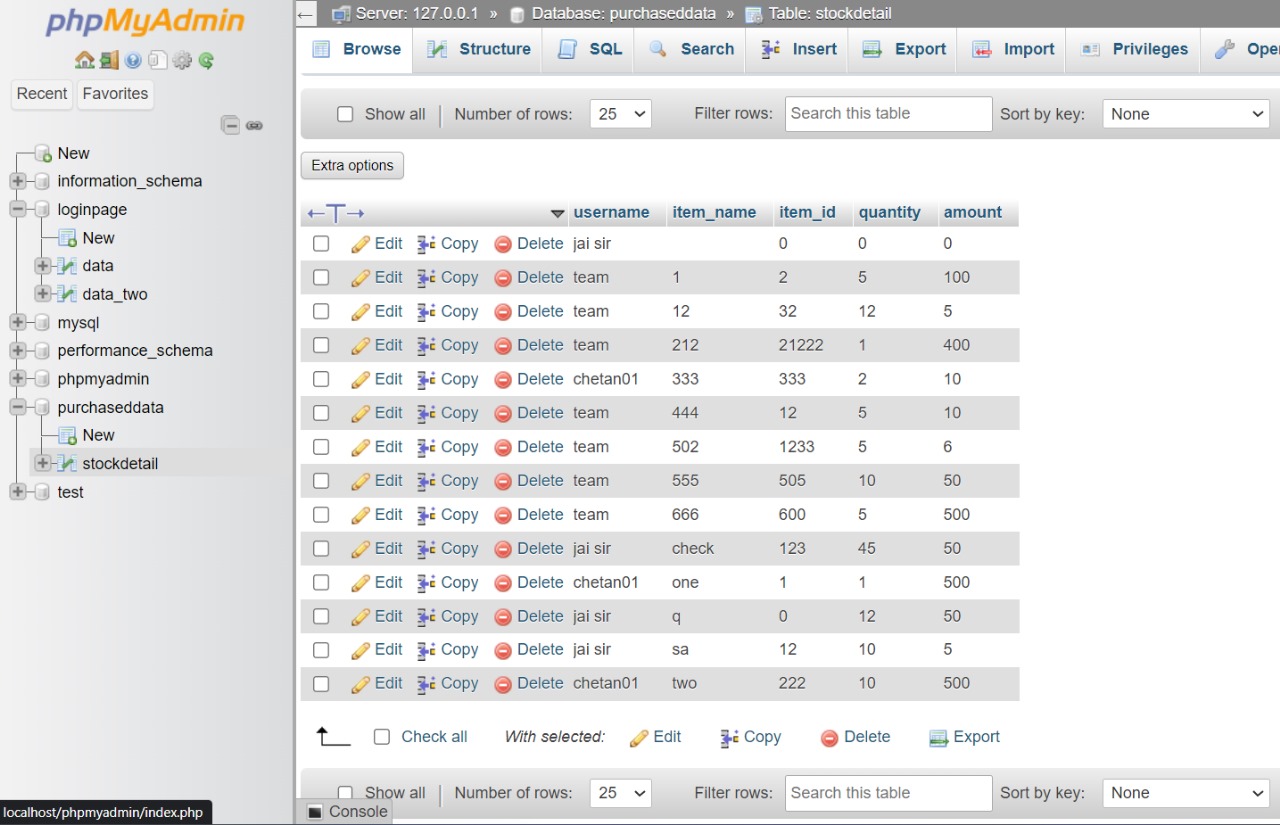
quantity: The quantity of the item in stock.

amount: The monetary value or cost associated with the item.

**9.b) Database:**

**Signup database:**

****

**Stock Details Database:**

**9.c) Pseudocode**

Step 1: Start

Step 2: Customer provides necessary information through GUI

Step 3: If input fields are not empty and Customer Login credentials are valid then

- Welcome the customer

- Display available products with prices

- Customer selects a product

Step 4: If the selected product exists then

- Admin verifies product availability

- If the product is available then

- Calculate total price

- Generate invoice with customer and product details

- Complete order placement

- Else, display a product unavailability message

Step 5: If the selected product does not exist, display an invalid product message

Step 6: If Customer Login credentials are invalid, display an invalid login credentials message

Step 7: Stop

**Login page pseudocode**

Step 0: Start

Initialize the system for processing customer orders.

Step 1: Customer Registration

Gather and input customer information.

Validate the entered information.

If valid, create a new customer record in the database.

Provide a success message for registration.

Step 2: Customer Login

Gather and input customer login credentials.

Validate the entered credentials.

If valid, verify customer credentials and welcome the customer.

Step 3: Customer Selects Product

Display available products.

Prompt the customer for the product ID they want to order.

Validate the entered product ID.

Retrieve product information from the database.

Return the selected product information.

Step 4: Admin Verifies Product

Retrieve product information from the customer's selection.

Verify product availability.

Print product information and request admin confirmation.

If admin confirms, return product information.

Step 5: Calculate Total Price

Calculate the total price based on the selected product's price.

Step 6: Generate Invoice

Create an invoice with customer information, product details, and total price.

Save the invoice for record-keeping.

Step 7: Customer Places Order

Prompt the customer to confirm their order.

If confirmed, consider the order placed.

If not, halt the process.

Step 8: Order Completion

Acknowledge the completion of the order process.

Provide any necessary closing messages or actions.

Step 9: Stop

**Pseudocode of Invoice**

Step 1: Customer Information

1.1 Prompt the customer to enter their name

1.2 Prompt the customer to enter their email address:

1.3 Prompt the customer to enter their shipping address:

Step 2: Product Information

2.1 Prompt the customer to select the product they want to order

2.2 Retrieve product information from the database:

Step 3: Calculate Total Price

3.1 Extract product price from product information:

3.2 Calculate applicable discounts

3.3 Calculate applicable taxes:

3.4 Calculate total price:

Step 4: Generate Invoice

4.1 Create an invoice object with customer information, product information, and

total price:

Step 5: Send Invoice to Customer

5.1 Send the invoice object to the customer's email address:

Step 6: Update Inventory

6.1 Reduce the stock quantity of the ordered product:

Step 7: Log Transaction

7.1 Create a transaction record in the database with the customer information, product information, and total price:

**Invoice Recording purchase transaction:**

Step 1: Get customer information

Prompt the customer to enter their name, email address, and shipping address.

Validate the customer's information.

Store the customer's information in the database.

Step 2: Get product information

Prompt the customer to select the product they want to order.

Retrieve the product information from the database.

Display the product information to the customer.

Step 3: Calculate total price

Get the product price from the product information.

Calculate any applicable discounts or taxes.

Calculate the total price.

Step 4: Process payment

Prompt the customer to enter their payment information.

Process the payment.

Step 5: Update inventory

Reduce the stock quantity of the ordered product.

Step 6: Generate invoice

Create an invoice object with the customer information, product information, and total price.

Save the invoice object to the database.

Step 7: Send invoice to customer

Send the invoice object to the customer's email address.

Step 8: Update order status

Update the order status to "paid".

**Admin panel pseudocode**

- Admin panel login

Step1: User enters username and password

Step2: System checks if username and password are valid

Step3: If valid, user is logged in

Step4: If invalid, user is prompted to try again

-Create user

Step1: Admin enters username, password, and email address for new user

Step2: System checks if username already exists

Step3: If username does not exist, new user is created

Step4: If username exists, admin is prompted to try again

- Modify user

Step1: Admin selects user to modify

Step2: Admin enters new username, password, or email address for user

Step3: System updates user information

Step4: User is notified of changes

- Delete user

Step1: Admin selects user to delete

Step2: System prompts admin to confirm deletion

Step3: If admin confirms, user is deleted

Step4: User's data is removed from the system

-Add product

Step1: Admin enters product name, price, and description

Step2: System checks if product name already exists

Step3:If product name does not exist, new product is added

Step4:If product name exists, admin is prompted to try again

-Remove product

Step1: Admin selects product to remove

Step2: System prompts admin to confirm removal

Step3: If admin confirms, product is removed

Step4: Product's data is removed from the system

-Update product

Step1: Admin selects product to update

Step2: Admin enters new product name, price, or description

Step3: System updates product information

Step4: Product is updated in the system

-Process billing

Step1: System retrieves all unpaid bills

Step2: For each unpaid bill, system processes payment

Step3: If payment is successful, bill is marked as paid

Step4; If payment is unsuccessful, bill remains unpaid

**7. Test Plan**

**a.) Functional, Performance, Stress tests etc.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Test Case ID** | **Steps** | **Input**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Remark** |
| 1. | Login\_01 | Open the login page | N/A | The login page is displayed. | The login page is displayed as expected. | Pass |
| 2. | Login\_02 | Enter a valid email and password. | Valid email and password. | User is successfully logged in, directed to the dashboard. | User successfully logged in. | Pass |
| 3. | Login\_03 | Enter an invalid email and valid password. | Invalid email and valid password. | Display an error message for incorrect email. | Error message displayed for incorrect email. | Pass |
| 4. | Login\_04 | Enter a valid email and an invalid password. | Valid email and invalid password. | Display an error message for incorrect password. | Error message displayed for incorrect password. | Pass |
| 5. | Login\_05 | Leave the email field blank and enter a valid password | Blank email and valid password. | Display an error message for a required email field | Error message displayed for required email field. | Pass |
| 6. | Login\_06 | Enter a vaild email and password blank. | Valid email and blank password. | Display an error message for required password. | Error message for invalid email format. | Pass |

**Functional Testing: Test Plan of login page**

**Test case table for Stock Detail:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Test Case ID** | **Steps** | **Input**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Remark** |
| 1. | Stock\_01 | Open the stock details page. | N/A | The stock details page is displayed. | The login page is displayed as expected. | Pass |
| 2. | Stock\_02 | Input valid item name, item id, quantity ,amount and total | Valid input data for item name, item id, quantity, amount, total. | The stock details are updated with the provided information. | Stock details are updated successfully. | Pass |
| 3. | Stock\_03 | Input valid item id. | Invalid item id. | Display an error message for invalid item id. | Error message displayed for invalid item id. | Pass |
| 4. | Stock\_04 | Input negative quantity. | Negative Quantity. | Display an error message for negative quantity. | Error message displayed for zero amount. | Pass |
| 5. | Stock\_05 | Input valid data and check the calculated data . | Valid input data for item id , quantity, amount. | Verify that the total is calculated correctly based input. | Total calculated correctly. | Pass |
| 6. | Stock\_06 | Input price with decimals and check total calculation. | Valid input data with a price containing decimals. | Verify that the total is calculated correctly for decimal prices. | Total shows wrong number after third decimal. | ail |

**Test case table for invoice generation:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Test Case ID** | **Steps** | **Input**  **Data** | **Expected**  **Result** | **Actual**  **Result** | **Remark** |
| 1. | Invoice\_01 | Open the invoice generation page. | N/A | The invoice generation page is displayed. | The invoice generation page is displayed as expected. | Pass |
| 2. | Invoice\_02 | Input valid customer name, invoice number, address and date. | Valid input data for item customer name, invoice number, address and date. | Customer information is populated in the invoice form. | Customer information form populated successfully. | Pass |
| 3. | Invoice\_03 | Input invalid invoice number. | Invalid invoice number. | Display an error message for invalid invoice number. | Error message displayed for invalid invoice number. | Pass |
| 4. | Invoice\_04 | Input future data as the invoice date. | Future date as the invoice date. | Display a warning message for a future invoice date. | Warning message displayed for a future date. | Pass |
| 5. | Invoice\_05 | Input invalid GST percentage. | Invalid GST percentage. | Display an error message for an invalid GST percentage. | Error message displayed for invalid GST percentage. | Pass |
| 6. | Invoice\_06 | Input negative quantity. | Negative quantity. | Display an error message for a negative quantity. | Error message displayed for zero rate. | Pass |

**10.a.(a)Functional Testing:**

Functional testing of your Quick Billing website involves evaluating its features and functionalities to ensure they meet the specified requirements.. Functional testing can be conducted for your Quick Billing project:

1. User Registration:

- Objective: Ensure users can register successfully on the website.

- Test Steps:

1. Open the registration page.

2. Input valid registration details.

3. Submit the registration form.

- Expected Result: User should be registered, and relevant data stored in the database.

- Actual Result: User is registered successfully, and data is stored as expected.

2. Product Management:

- Objective: Verify the functionality to add, update, and remove products.

- Test Steps:

1. Navigate to the product management section.

2. Add a new product.

3. Update product details.

4. Remove a product.

- Expected Result: Products are added, updated, and removed accurately.

- Actual Result: Product management functions correctly.

3. Order Processing:

- Objective: Test the end-to-end order processing workflow.

- Test Steps:

1. Select products and add them to the cart.

2. Verify the cart contents.

3. Place an order.

- Expected Result: Order is placed successfully, and data is recorded accurately.

- Actual Result: Order processing functions as expected.

4. Invoice Generation:

- Objective: Validate the generation of invoices for completed orders.

- Test Steps:

1. Access the invoice generation page.

2. Input valid data for customer and product details.

3. Generate an invoice.

- Expected Result: Invoices are generated with correct details.

- Actual Result: Invoice generation is successful, and details are accurate.

**10.a.(b)Performance Testing:**

Performance testing is essential to ensure that your Quick Billing website can handle various loads and provides a satisfactory user experience under different conditions. Performance Testing for Quick Billing Website:

Objective: Evaluate Response Time

Test Scenario: Assess the response time of critical functions such as page loading, form submission, and data retrieval.

Test Steps:

Measure the time taken for key pages to load.

Analyze the time required for form submissions.

Evaluate the speed of data retrieval from the database.

Expected Result: Pages load within an acceptable time frame, and form submissions are prompt.

Actual Result: Response times are measured and meet or exceed acceptable benchmarks.

Objective: Database Query Optimization

Test Scenario: Evaluate the efficiency of database queries to ensure that data retrieval is optimized.

Test Steps:

Execute queries for various scenarios (e.g., product lookup, order history).

Measure the time taken for data retrieval.

Optimize queries for better performance.

Expected Result: Database queries should execute efficiently, and optimizations, if necessary, should be implemented.

Actual Result: Database queries are analyzed and optimized for improved performance.

**10.a.(c)Stress Testing:**

Stress testing is a type of performance testing that involves evaluating the stability and robustness of your website under extreme conditions or loads. The goal is to determine how well the system can handle increased stress, such as a surge in user activity or a high volume of transactions.

Stress Testing for Quick Billing Website:

Objective: Test Concurrent Transactions

Test Scenario: Simulate a high volume of concurrent transactions to assess the system's ability to handle simultaneous user actions.

Test Steps:

Execute a large number of transactions simultaneously.

Monitor transaction processing times and system responsiveness.

Expected Result: The system should process transactions efficiently, even under high concurrency.

Actual Result: Evaluate how well the system maintains transaction processing times under stress.

Objective: Validate Database Performance

Test Scenario: Stress the database by generating a significant volume of read and write operations to evaluate its performance under load.

Test Steps:

Execute a high volume of database transactions.

Monitor database response times and query execution.

Expected Result: The database should handle the increased load efficiently, with optimized query execution.

Actual Result: Assess how well the database performs under stress and optimize queries if necessary.

**11. Implementation/Conversion Plan:**

Objective:

The objective of this conversion plan is to seamlessly transition from the current manual record-keeping system to the Quick Billing website. The aim is to enhance efficiency, reduce administrative burden, and provide insightful statistical overviews for shopkeepers and businesses.

Project Initiation:

Scope Definition:

The scope includes developing an automated billing system that streamlines purchase and sales records, reducing errors and improving overall efficiency.

Project Team Formation:

Assemble a dedicated project team comprising frontend and backend developers, database administrators, UI/UX designers, and a project manager.

Timeline and Budgeting:

Establish a realistic timeline for development, testing, and deployment phases.

Allocate a budget for development resources, software tools, and promotional activities.

System Analysis:

Requirement Gathering:

Gather detailed requirements from shopkeepers, potential users to ensure the system addresses their specific needs.

Technology Stack Selection:

Choose appropriate technologies for frontend (HTML, CSS, JavaScript), backend (PHP), and database (MySQL) based on project requirements.

Database Design and Setup:

Database Schema Finalization:

Finalize a comprehensive database schema that accommodates purchase and sales records, user data, and customizable features.

Database Environment Setup:

Set up a secure and efficient database environment, ensuring proper connectivity and data integrity.

Development:

Frontend Development:

Develop an intuitive and user-friendly interface, incorporating features such as bill recording, statistical overviews, and customization options.

Backend Development:

Implement robust server-side logic for user authentication, record-keeping, and seamless integration with the database.

Testing:

Unit and Integration Testing:

Conduct thorough testing of individual components and integrated functionalities to ensure robustness.

User Feedback Collection:

Collect user feedback post-launch to further optimize the system based on real-world usage.

**12.Project Legacy**

**(a) Current status of project**

As of the current status, our billing website project is in an active and dynamic phase, with the development and implementation processes in full swing. The team is diligently working on refining features, enhancing user interfaces, and addressing any remaining bugs or issues to ensure a seamless and efficient billing experience for our users. Regular collaboration and feedback loops with stakeholders are ongoing, allowing us to incorporate valuable insights and make real-time adjustments. The project remains on track in terms of timelines and milestones, and the commitment to delivering a high-quality billing system that meets the needs of our users and organizational objectives remains steadfast. We are excited about the progress made thus far and are dedicated to achieving a successful and impactful outcome as we move forward in the development lifecycle.

**(b) Remaining Areas of Concern**

While significant strides have been made in the development of our billing system website, there are still areas of concern that warrant attention, specifically the completion of the back-end functionality for the invoice page. The team is currently focused on finalizing the intricate details of the invoicing module, ensuring that it seamlessly integrates with the front-end components and aligns with our design specifications. Challenges related to data synchronization, real-time updates, and robust data validation are being actively addressed to guarantee the accuracy and reliability of the invoice generation process. Additionally, efforts are underway to optimize the back-end architecture for scalability and performance. We are committed to overcoming these remaining challenges swiftly and efficiently, maintaining open communication channels within the team to collaborate on innovative solutions. This targeted approach ensures that our billing system will not only meet but exceed the expectations of our users when it comes to invoicing functionality. We appreciate the dedication of our team in resolving these concerns and are confident in achieving a robust and comprehensive billing solution.

**(c)Technical Lessons**

1. User Experience (UX) Design:

Prioritizing a user-friendly interface is key to customer satisfaction. Regular user testing and feedback loops during the design phase help create an intuitive and efficient billing experience.

2. Error Handling and Logging:

Robust error handling mechanisms and detailed logging are critical for identifying and resolving issues promptly. Proper logging aids in troubleshooting and debugging during both development and production phases.

3. Cross-Browser Compatibility:

Ensuring cross-browser compatibility during development prevents issues related to inconsistent rendering and functionality across different web browsers, enhancing the user experience.

4. Scalable Database Design:

Designing the database schema with scalability in mind, including sharding, partitioning, or replication strategies, prepares the system for growth in data volume and user base.

5. Database Indexing and Query Optimization:

Implementing effective database indexing and optimizing queries enhances database performance. Efficient database operations are crucial for maintaining responsiveness in large-scale billing systems.

6. Load Testing and Performance Optimization:

Conducting load testing early in the development process helps identify potential bottlenecks. Optimizing performance based on test results ensures the system can handle expected user loads.

7. Backup and Disaster Recovery:

Implementing regular data backups and establishing a robust disaster recovery plan ensures data integrity and minimizes downtime in the event of system failures or data loss.

**(c)Managerial Lessons**

1. Clear Requirements Definition

A thorough understanding of user requirements is crucial. Clear and detailed requirements ensure that the development team builds a billing system that aligns with the organization's needs and goals.

2. Effective Communication

Establishing effective communication channels between team members, stakeholders, and users is essential. Regular updates, feedback sessions, and transparent communication foster collaboration and mitigate misunderstandings.

3. User Involvement:

Involving end-users in the development process through regular feedback sessions ensures that the billing system meets their expectations and addresses their pain points effectively.

4. Resource Allocation:

Efficiently allocating resources, both human and technological, based on project priorities and timelines, is critical for meeting milestones and deadlines.

5. Continuous Improvement

Embracing a culture of continuous improvement encourages the team to learn from experiences, identify areas for enhancement, and implement lessons learned in subsequent phases or projects.

**(d)Future Recommendations**

1.Continuous Innovation and Feature Enhancement:

- Foster a culture of perpetual innovation within the development team, encouraging exploration and implementation of features that align with emerging technologies. Establish agile development practices to swiftly adapt the billing system to changing technological landscapes.

2. Integration with Emerging Technologies:

- Proactively seek collaborations with technology providers specializing in emerging technologies. Develop an adaptable architecture that allows seamless integration of artificial intelligence, machine learning, and blockchain as these technologies continue to evolve.

3.Enhanced Mobile Experience:

- Prioritize the adaptability of the billing system for evolving mobile technologies. Invest in progressive web app development, ensuring the system's responsiveness and optimal performance across a spectrum of mobile devices and platforms.

4. Real-Time Analytics and Reporting:

- Invest in future-proof analytics and reporting tools that leverage technologies like edge computing and advanced data processing. Ensure that the system architecture supports the integration of upcoming advancements in real-time data analytics.

5.Personalization and Customization:

- Develop a modular and extensible system architecture that allows for easy integration of personalized features. Anticipate advancements in user customization preferences and interface with future technologies that enhance personalization.

6. User Feedback Mechanism:

- Establish an adaptive and user-friendly feedback mechanism within the billing system. Integrate sentiment analysis and natural language processing to extract meaningful insights from user feedback, ensuring continuous improvement aligned with future user expectations.

**13. Bibliography**

* Post, J. (2023) *Billing tips for B2B businesses*, *Business News Daily*. Available at: https://www.businessnewsdaily.com/10564-best-b2b-billing-practices.html (Accessed: 29 July 2023).
* India, Govt. of (2023) *© 2020. Central Board of Indirect Taxes and Customs, Government of India, New Delhi, India*, *Goods & Service Tax, CBIC, Government of India :: GST goods and services rates*. Available at: https://cbic-gst.gov.in/gst-goods-services-rates.html (Accessed: 29 July 2023).
* .BILL, LLC. (2023) *Online bill payment and receivables software*, *BILL*. Available at: https://www.bill.com/home-test-v1 (Accessed: 29 July 2023).
* Inc., © 2022 Khatabook (2022) *Khatabook*, *Khatabook.com*. Available athttps://khatabook.com/ (Accessed: 03 October 2023).
* Gupta, A. (2022) *How will the changes made to the e-invoicing system impact the business?*, *Times of India Blog*. Available at: https://timesofindia.indiatimes.com/blogs/voices/how-will-the-changes-made-to-the-e-invoicing-system-impact-the-business/ (Accessed: 03 November 2023).
* Blog, A. (2023) *GST’s next chapter: The Future of Tax System in India*, *Online Accounting Software in India*. Available at: https://ankpal.com/blogs/gst-next-chapter-future-of-tax-system (Accessed: 03 November 2023).
* Clap, B. (2023) *GST billing software setup: A step-by-step guide for Indian businesses*, *BillClap*. Available at: https://www.billclap.com/blog/gst-billing-software-setup:-a-step-by-step-guide-for-indian-businesses (Accessed: 13 October 2023).

14. Source Code

Official\_Website.html(Main page of Quick Billing System)

Source Code:  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Billing System</title>

<link rel="stylesheet" href="Official\_Website.css">

<!-- -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<link rel="icon" type="image/x-icon" href="favicon.png">

<script src=https://code.jquery.com/jquery-3.6.3.min.js></script>

<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

<link rel="stylesheet" href=”https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css” />

</head>

<body>

<nav class="navbar">

<div class="container">

<a href="Official\_Website.html"><img src="Logo1.png" alt="Logo" class="logo"></a>

<div class="login-signup1">

<a href="Official\_Website.html" class="login">

<i class="fa fa-user"></i>&nbsp;&nbsp;REGISTER</a>

</div>

<div class="sidemenu" style="margin-left:92%;margin-top:-9%">

<i class="fas fa-bars"></i>

</div>

<ul class="nav-links">

<li><a href="Official\_Website.html">Home</a></li>

<li><a href="Inventory\_Page.html">Inventory</a></li>

<li><a href="#f1">Contact Us</a></li>

<li><a href="#f1">About Us</a></li>

</ul>

<div class="login-signup" id="regis">

<a href="loginofficial.html" class="login">

<i class="fa fa-user"></i>&nbsp;REGISTER

</a>

</div>

</div>

<ul class="side-links">

<li><a href="#">Home</a></li>

<li><a href="#">Inventory</a></li>

<li><a href="#">Contact Us</a></li>

<li><a href="#">About Us</a></li>

</ul>

</nav>

<div class="div2" style="width:85vw;height: 70vh; margin-top:5%">

<div class="text"> <br> <br>

<h1 class="animated-text">Welcome To The<br> <span style="color:#3473d1">Billing System</span></h1>

<br><a href="#features" class="custom-button">Start Discover Now</a>

<!--Scroll from Start Discover Now Button to features-->

</div>

<div class="photo">

<img src="side-image2.jpg" style="width:50vw;height:70vh; ">

</div>

</div>

<a href="#"><button class="down"><i class="fa-solid fa-circle-arrow-up"></i></button></a>

<!-- Below Button is only for my reference -->

<a href="#footer-bottom"><button class="up"><i class="fa-solid fa-circle-arrow-down"></i></button></a>

</div>

<div class="twpage" id="twpage" style="background-color:#030844;top:0%">

<div class="heading"><br>

<h1> The New Standard For Bill Payment</h1>

</div>

<div class="row">

<div class="box" id="box2" Style="top:0;">

<h2>Easy To Use</h2>

<p>Easily understandable and seamlessly intuitive interface. 🌟🖥

</p><br><br>

</div>

<div class="box" id="box2">

<h2>Efficient</h2>

<p>🕒💪 Save your Valuable Time and Effort 💡🚀</p><br><br><br>

</div>

<div class="box" id="box3" Style="border-right:#150062;border-bottom:1 px solid #030844">

<h2>Precision and Accuracy</h2>

<p>Precision-driven calculations and meticulously kept records. ✨📊</p><br><br><br>

</div>

<section id="features"></section> <!--Scroll from Start Discover Now Button to features-->

</div>

</div>

<br><br>

<h1 style="text-align: center;" class="features">FEATURES</h1>

<div class="parent" id="parent">

<div class="child">

<img src="feature\_1.jpg" alt="Image 1">

<div class="text">

<h2>Feature 1</h2>

<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Rerum sequi quibusdam non incidunt sed

quasi

dolorem! Consequuntur reprehenderit, facere repellat unde nemo expedita accusantium tempora aliquam

vel

dolores. Nulla, iste.</p>

</div>

</div><br>

<div class="child">

<div class="text">

<h2>Feature 2</h2>

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Maiores qui ad, necessitatibus eligendi

illo

cumque temporibus nisi fugiat hic facilis velit atque dolorum aspernatur, quas obcaecati? Error

harum et

dolore.</p>

</div>

<img src="feature\_2.jpg " alt="Image 2">

</div><br>

<div class="child">

<img src="feature\_1.jpg" alt="Image 3">

<div class="text">

<h2>Feature 3</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui similique vel aperiam aliquam

necessitatibus

neque! Architecto, deserunt ratione, autem aliquam illo suscipit exercitationem ex eveniet,

voluptate

corporis blanditiis beatae dolore!

</p>

</div><br>

</div>

<div class="child">

<div class="text">

<h2>Feature 4</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi doloremque consequuntur voluptates

explicabo, exercitationem, aut vitae architecto quod voluptas cumque voluptatem fugit dignissimos

quam

vero sed accusamus blanditiis officia? Minus.</p>

</div>

<img src="feature\_4.jpg" alt="Image 4">

</div>

</div>

<section id="footer-bottom"></section>

<footer class="footer" id="f1">

<div class="container3">

<div class="row3">

<div class="footer-col">

<h4>Quick Billing</h4>

<ul>

<li><a href="#parent">Features</a></li>

<li><a href="#">How it works?</a></li>

<!-- <li><a href="#">Specification</a></li> -->

<li><a href="team2.html">Our Team</a></li>

</ul>

</div>

<div class="footer-col">

<h4>Get Help</h4>

<ul>

<li><a href="faq.html">FAQ</a></li>

<li><a href="#">Contactus</a></li>

<!-- <li><a href="#">Sales</a></li>

<li><a href="#">Returns</a></li>

<li><a href="#">Payment</a></li> -->

</ul>

</div>

<div class="footer-col">

<h4>Online Activity</h4>

<ul>

<li><a href="#">Watch</a></li>

<li><a href="#">Blog</a></li>

<!-- <li><a href="#">Register</a></li>

<li><a href="#">GST</a></li> -->

</ul>

</div>

<div class="footer-col">

<h4>Follow Us</h4>

<div class="social-links">

<div class="facebook"><a href="#" target="\_blank"><i class="fab fa-facebook-f"></i></a></div>

<div class="twitter"><a href="https://twitter.com/AashutoshBajaj2" target="\_blank"><i

class="fab fa-twitter"></i></a></div>

<div><a href="" target="\_blank"><i class="fab fa-instagram"></i></a></div>

<!-- <div><a href="" target="\_blank"><i class="fab fa-youtube"></i></a></div> -->

</div>

</div>

</div>

</div>

</div>

</footer>

<script>

$(document).ready(function () {

$('.sidemenu').click(function () {

$('.nav-links').toggle();

});

});

</script>

</body>

</html>